Autopsy of a BYOK solution

Martin Rupp
SCIENTIFIC AND COMPUTER DEVELOPMENT SCD LTD

If you're using AWS, Google Cloud, MS Azure, or any other cloud you are probably familiar with the following concepts: KMS and BYOK.

Introduction to KMS

KMS means Key Management Service. Usually, it is a secure and resilient service based on hardware security modules that have been validated under FIPS 140-2. Each cloud provider has its own KMS. As an example, we listed below the KMS solution for the biggest cloud provider (as of 2022):

Cloud	KMS
AWS	AWS KMS
Google Cloud	Cloud KMS
MS Azure	Key Vault
IBM Cloud	IBM Key Protect
Alibaba Cloud	Alibaba Cloud Key

	Management Service
Oracle Cloud	Oracle Cloud KMS

Here are a few benefits of using a KMS:

- A KMS is fully managed;
- A KMS is centralized key management;
- A KMS is interoperable with other Services from the same cloud provider;
- A KMS is Secure and Compliant with several norms;
- A KMS provides logging and auditing;
- Possibility to *lease* temporarily the keys to the KMS (for instance supported by AWS …)
- A KMS (usually) an API.

However, using a KMS presents several problems. The biggest of them is that there is a risk of a vendor lock-in.

A KMS, which is a native key management solution, tends to lock the user because moving the data to another cloud provider is impossible without deciphering the data first.

Introduction to BYOK

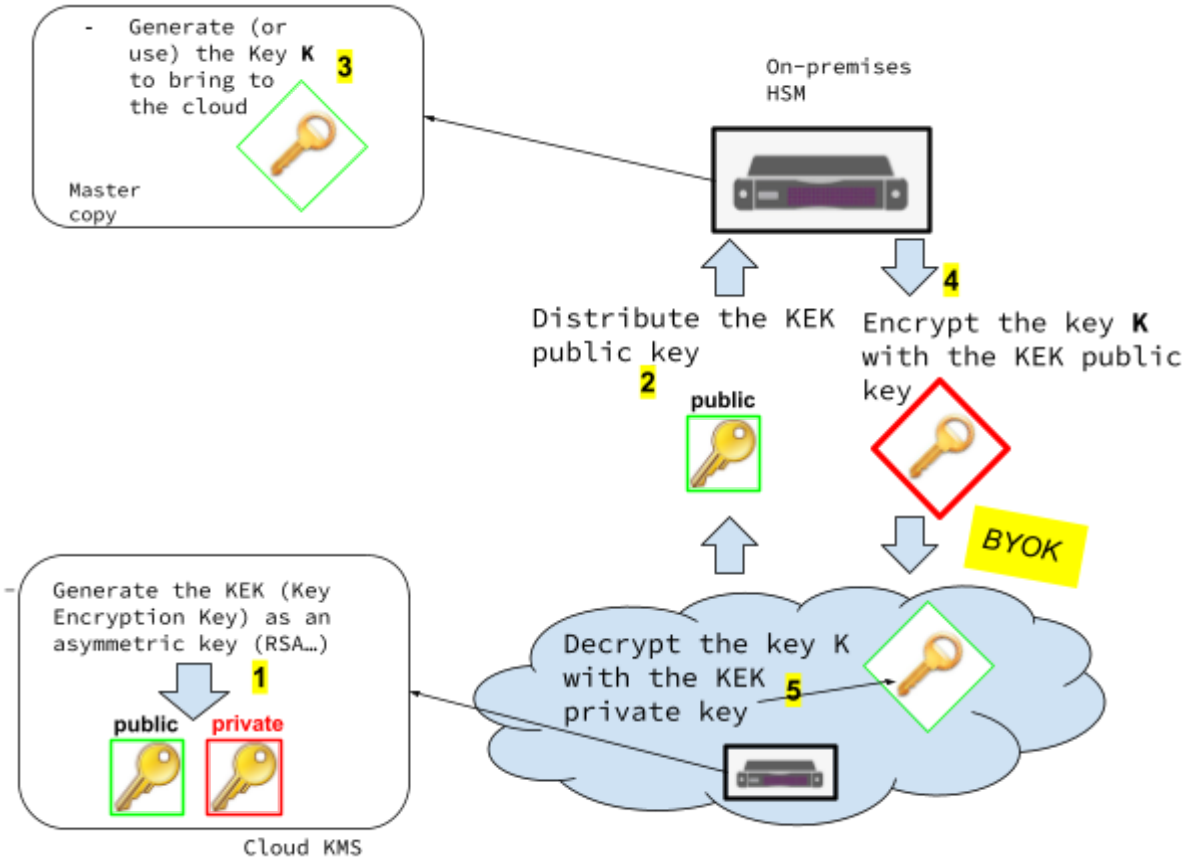
BYOK means 'bring your key'. It's a feature supported by several of the cloud KMS. The idea is that the key is first generated/stored in an external structure. Generally, an HSM and then it is exported to the KMS. The usual way to load the external key into the KMS is via asymmetric cryptography.

Example of a BYOK procedure

As an example, we detail an example of a typical BYOK procedure.

1. *Generate the encryption key (KEK) in the KMS.*
2. *Retrieve the public key of the KEK from the KMS.*
3. *Using an HSM vendor-provided BYOK tool - Import the KEK into the target HSM and export the Target Key protected by the KEK.*
4. *Import the protected Target Key to the KMS.*
5. *Decipher the Protected Target Key with the private key part of the KEK.*

The principle is simple and several key block formats can generally be used to securely transfer the key to the cloud KMS.



This is an example of a BYOK procedure. There are several other ways to do it depending on what is supported by the cloud KMS. Potentially popular key block formats such as Atalla, Thales, or TR-31 key block formats may be used.

A specific BYOK program is needed to perform the operation.

How BYOK works for AWS

In what follows we will detail the BYOK procedure for AWS

The first step is for the user to create AWS KMS keys (KMS key) with its own key material.

A KMS key contains key material as well as key identifiers and other metadata (in fact this KMS key format is equivalent to a key block format)

AWS KMS doesn’t expose the exact format it uses for encryption and decryption (see disadvantages of BYOK in the final chapter of this document) so it is not guaranteed that a user can decipher data stored in the Amazon cloud outside the Amazon infrastructure, even if the user possesses the right key.

Here are the benefits for users of generating their own key and exporting it to the KMS:

- This allows users to prove that they generated the key material using a source of entropy that meets their own requirements;
- They can use key material from their own infrastructure with AWS services, and use AWS KMS to manage the lifecycle of that key material;
- Users can set an expiration time for the key material (in AWS);
- Users own the original copy of their own key material, and keep it outside of AWS for additional durability and disaster recovery during the complete key lifecycle.

Detailed procedure of a BYOK for AWS

1) The first step is to create in the KMS of AWS a KMS key containing no key material. This key will be marked as a key for symmetric encryption and which can only receive key material from external sources (e.g. a BYOK key).
2) The AWS KMS will then require the user to download a public key as well as extra binary data named the import token, which provides additional security.
3) A key material must be created by ciphering the key material with the public key.
4) The encrypted key material must then be uploaded together with the import token data to the KMS. The KMS then automatically processes the upload and ultimately will fill the KMS key with the key material if all the steps have been correctly fulfilled.

All these steps must usually be done via a specific BYOK program from the user's own key management software.

Limitations of BYOK

Limitation

BYOK has several important limitations and caveats. One of them is that the cloud provider will store and manage a copy of the keys so, in fact, with BYOK, the keys are concretely shared between the cloud and their clients. BYOK should be called SYOK for Share Your Own Key. Because of the BYOK procedure, there can be little to do if a key is stolen.

BYOK also doesn't entirely solve the vendor lock-in problem. Because even if technically the user has the keys needed to decipher the data stored in the cloud, they may not know the full details of the encryption parameters used by the cloud provider and the same cloud provider may be at no obligation to share them with their client, thus making the migration to another cloud provider quite difficult.

CYOK and HYOK

CYOK

"Control Your Own Keys" or CYOK is a cloud key management concept where the user has control over the keys that are used by the cloud provider for encrypting and deciphering their data. This control usually always involves immediate revocation of the keys at any time by the user. CYOK can be used in combination with BYOK or not. In both cases, the data encryption keys are stored and used by the cloud provider.

HYOK

"Hold Your Own Keys" or HYOK is currently the most secure cloud key management scenario. In such a case, the user retains the encryption keys on their premises. They cipher and decipher the data in their infrastructure as well as store already ciphered data in the cloud. Usually, HYOK is achieved via a specific 'driver' between the user's infrastructure and the cloud which allows the cloud to access 'external' ciphering and deciphering.
HYOK allows finer granularity than BYOK.

Complexity of CYOK and HYOK compared to BYOK

These two solutions, and especially HYOK, are much more complex and more difficult to set up and use than BYOK. For many companies, BYOK is the ideal solution as it is a balance between simplicity of use and security.

Conclusion: why using BYOK for AWS is a smart move

Using BYOK is in general generally a smart move for cloud KMS users. It allows users to make sure the encryption material used in a cloud KMS is compliant with their own requirements and it allows finer control of the cloud KMS encryption. BYOK for AWS KMS is, overall, a simple procedure that doesn't necessitate a lot of knowledge in cryptography and can be achieved via specific BYOK services. Additionally, this combines with the power of the AWS Key Management System:

- Access control of the KMS keys using various policies
- Ability to tag the KMS keys
- Automatic rotation of the keys
- Encryption of data of course but also signature and verification of messages
- Multi-region keys
- Many other cryptographic features